B. AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph beginning at page 1, line 16, with the following paragraph:

Application S. N. 07/352,074 ~~(IBM internal docket number AT9-89-035)~~ filed May 15, 1989 in the name of R. J. Archon, now abandoned, ~~for "An Initial Program Load (IPL) Based On An Object Abstraction For A Data Processing System",~~ hereby incorporated by reference.

Please amend the paragraph beginning at page 1, line 21, with the following paragraph:

Application S. N. 07/352,571 ~~(IBM internal docket number AT9-89-031)~~ filed May 15, 1989 in the name of R. A. Fabbio, now abandoned, ~~for "An Open System Management Architecture For A Data Processing System",~~ hereby incorporated by reference.

Please amend the paragraph beginning at page 1, line 26, with the following paragraph:

Application S. N. 07/352,081 ~~(IBM internal docket number AT9-89-038)~~ filed May 15, 1989 in the name of R. A. Fabbio, now abandoned, ~~for "Access Control Policies For An Object Oriented Database"~~, hereby incorporated by reference.

Please amend the paragraph beginning at page 7, line 9, with the following paragraph:

A distinction is made between interface data objects and system resource data objects, otherwise referred to as application data objects. System resource objects are described in more detail with reference to Serial Number 07/352,074 ~~(IBM internal docket number AT9-89-035)~~, filed May 15, 1989 in the name of R. J. Archon, now abandoned, ~~for "An Initial Program (IPL) Based on the Object Abstraction for a Data Processing system"~~, hereby incorporated by reference; and Serial Number 07/352,571 ~~(IBM internal docket number AT9089-031)~~, filed May 15, 1989, in the name of R. A. Fabbio, now abandoned, ~~for "An Open system Management Architecture For A Data Processing System"~~, hereby incorporated by reference. System resource data objects define the workings of the system, are installed with the software (application or system) that control them, are read/write to the application layer, and are of no direct interest to the interface tool.

Please amend the paragraph beginning at page 13, line 20, with the following paragraph:

The menu object class 200 is shown in Fig. 2. The id 202 is the name of the object. This ~~filed~~ field holds information sufficient to provide a search key by means of which this object is pointed to by the previous level object, thereby determining this

object's participation with other objects in a group. This field may be externalized as a fast path id.

Please amend the paragraph beginning at page 13, line 27, with the following paragraph:

The id_seq_num 203 indicates the order in which the objects of the menu will be displayed. The next_id 204 indicates the id of the next object for which to search if this item is selected. The field text 205 describes the task to be presented to the user. The text_msg_file 206 is the message facilities catalogue for the string, text. The text_msg_set field 207 is the message facilities set id for the string, text. Set ids are assigned by message services. The text_msg_id field 208 is the message facilities message id for the string, name. The next_type ~~filed~~ field 209 indicates the type of the next object if this item is selected. Allowed values are "menu", "name", "dialogue", and "info". The alias field 210 indicates whether or not this menu object is a fast path pointer to another menu object. Allowed values are "yes" and "no". For a menu object which is an alias, the following convention applies: the objects next_id 204 and id_seq_num 203 will be used to find the menu object which will point to the menu desired. The help_msg_id 211 indicates the tag number for the contextual help for the object.

Please amend the paragraph beginning at page 15, line 23, with the following paragraph:

To create a dialogue on the screen, the interface tool searches for keyed data objects (the key having been supplied from the previous object's "next" field), chooses appropriate representations for the data objects (according to the current metaphor and/or NLS environment), orders the data objects according to their self-stated relationships to other possible objects, and prepares to present ~~then~~ them to the user. However, some aspects of the data objects (such as current values) may be missing. At this time, the interface tool invokes application layer functions to discover current configuration values. The names of these functions, as well, are contained within the dialogue data objects.

Please amend the paragraph beginning at page 21, line 17, with the following paragraph:

The cmd_to_list_mode 313 indicates how much of an item selected from a list should be used. Allowed values are sm_first_field and sm_all_fields. If the command returns a range rather than a list, the value of this field should be sm_range. ~~ranges~~ Ranges are not selectable, but are for information only. The cmd_to_list 312 indicates the operating system command, if one exists, which supplies a list of candidates for the value field. The field cmd_to_list_postfix 323 indicates the postfix to be used with the cmd_to_list. There are three allowed values; the first may be used only when there is a name selector associated with the dialogue. The sm_postfix_raw indicates that the name

entered into the name selector panel is to be used with the command. The value sm_postfix_cooked indicates the ~~the~~ output from the name selector's cmd_to_classify should be used with the command. The value sm_postfix_empty indicates that there is no postfix for the command. When this object is used for a name selector, this field is reserved and should be set to null.

Please amend the paragraph beginning at page 22, line 9, with the following paragraph:

The field disp_values 315 provides the current, default, or allowed values which are to be presented to the user at the beginning of a dialogue. This field may be initialized when the object is developed or discovered at runtime by the cmd_to_discover. The values_msg_file 316 is the message facilities catalogue for the string, ~~disp_values. If~~ disp_values, if these values are initialized when the object is developed. The values_msg_set field 317 is the message facilities set id for the string, disp_values, if these values are initialized at development time. The field values_msg_id 318 is the message facilities message id for the string, disp_values, if these values are initialized when the object is developed.

Please amend the paragraph beginning at page 46, line 10, with the following paragraph:

The various logical frame presentations that may be accessible by several system administrators having various system authority can be controlled by applying access control policies to the interface objects. The method of applying access control policies to objects is further described in Application S. N. 07/352,081 ~~(IBM internal docket number AT9-89-038)~~ filed May 15, 1989 in the name of R. A. Fabbio, now abandoned, ~~for "Access Control Policies For An Object Oriented Database"~~, hereby incorporated by reference. These access control policies can be applied to each interface object. A menu interface tool is operating in behalf if a user whose credentials do not meet those that are in the access control list attached to the interface object. Each of the various interface objects are stored at only one location in the object database, and are assigned the appropriate access controls such that the various permutations of administrative views are dictated by the access control policy for the objects within the view. For example, administrator A may have read, write, and execute privileges for the user interface objects which present the configurable domain of TCP/IP, devices, and users, while administrator B may have read, write, and execute privileges for the user interface objects which present the configurable domains of TCP/IP, SNA, devices, users, and systems. Therefore, it is possible to tailor the menus, dialogs, and options, that each administrator interacts with by setting the appropriate access control policies on the various objects which define the user interface.